Amendment to the Claims:

Without prejudice, this listing of the claims replaces all prior versions and listings of the claims in the present application:

Listing of Claims:

1. (Currently Amended) A method of managing visibility of GUI components in an application comprising:

providing a user interface of a visibility manager for selection, for each of a plurality of profiles, of one of a plurality of visibility states for each of at least a subset of the GUI components; and

for displaying a user interface screen of the application according to an applied one of the plurality of profiles:

initializing the application, wherein the initializing includes:

starting the application; and

building the user interface screen of the application with all of the GUI components set as visible;

invoking the visibility manager to:

determine, based on the selected visibility states of the applied profile, which of the GUI components of the built user interface screen are to be set as not visible; and

revise the built user interface screen based on the determination; and

displaying [[a]] the revised user interface screen of the application ~~wherein display of the GUI components are determined by the visibility manager based on the selected visibility states of an applied one of the plurality of profiles~~.

2. (Previously Presented) The method as recited in claim 1 wherein invoking the visibility manager comprises:

reading the plurality of profiles;

processing the plurality of profiles;

reading and processing a user configuration based on the plurality of profiles; and

activating the applied profile.

3-4. (Canceled).

5. (Previously Presented) The method as recited in claim 2 wherein activating the applied profile comprises:

selecting an identification of a particular GUI component;

locating the identification in a mapping table;

checking a state of the particular GUI component;

comparing the state to the applied profile;

changing the state if not in agreement with the applied profile; and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

6. (Original) The method as recited in claim 5 wherein the state is visible or not visible.

7. (Previously Presented) The method as recited in claim 5 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to objects of an object-oriented and platform independent programming language.

8. (Currently Amended) The method as recited in claim 1 further comprising:

invoking the visibility manager by:

reading the plurality of profiles,

processing the plurality of profiles,

reading and processing a user configuration based on the plurality of profiles, and

activating the applied profile by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table,

checking a state of the particular GUI component,

changing the state if not in agreement with the applied profile, and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components~~, revising the user interface based on the applied profile, and displaying the user interface; initializing the application by: starting the application, building the user~~

~~interface with all of the GUI components visible, and calling the visibility manager after the applied profile is activated~~.

9. (Currently Amended) A system for managing visibility of GUI components in an application comprising:

a processor configured to provide:

a user interface module of the application~~, including the GUI components~~;

a visibility manager ~~that determines which GUI components are visible~~; and

a user interface of the visibility manager via which to receive, for one or more profiles, input of respective selections of visibility states of at least a subset of the GUI components; ~~, the visibility manager determining which GUI components are visible based on the selections~~

wherein, for display of a single user interface screen of the application according to an applied one of the profiles:

the user interface module of the application is configured to initially build the user interface screen with all of the GUI components set as visible and subsequently call the visibility manager;

the visibility manager is configured to, in response to the call:

determine, based on the selected visibility states of the at least a subset of the GUI components, which of the GUI components of the built user interface screen are to be set as not visible;

revise the built user interface screen based on the determination; and

provide the revised user interface screen to the user interface module; and

the user interface module is configured to display the revised user interface screen.

10. (Previously Presented) The system as recited in claim 9, wherein

the visibility manager to determine which GUI components to display by, and the visibility manager further to:

read the one or more profiles upon initialization of the application;

process the one or more profiles;

read and process a user configuration based on the one or more profiles; and

activate a particular profile of the one or more profiles.

11. (Canceled).

12. (Original) The system as recited in claim 10 wherein the visibility manager to:

select an identification of a particular GUI component;

locate the identification in a mapping table;

check a state of the particular GUI component;

compare the state to the particular profile of the one or more profiles; and

change the state if not in agreement with the particular profile of the one or more profiles.

13. (Original) The system as recited in claim 12 wherein activating the particular profile of the one or more profiles further comprises repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

14. (Original) The system as recited in claim 12 wherein the state is visible or not visible.

15. (Previously Presented) The system as recited in claim 12 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to objects of an object-oriented and platform independent programming language.

16. (Currently Amended) The system as recited in claim 9 further wherein

the visibility manager to determine which GUI components to display by, and the visibility manager to:

read the one or more profiles upon initialization of the application,

process the one or more profiles,

read and process a user configuration based on the one or more profiles, and

activate a particular profile of the one or more profiles by:

select an identification of a particular GUI component,

locate the identification in a mapping table,

check a state of the particular GUI component,

compare the state to the particular profile of the one or more profiles, and

change the state if not in agreement with the particular profile of the one or more profiles~~, start the application, build the user interface with all of the GUI components visible, call the visibility manager after the particular profile of the one or more profiles is activated, revise the user interface based on the particular profile of the one or more profiles, and display the user interface~~.

17. (Currently Amended) A system comprising:

a processor configured to:

use a visibility manager data structure in managing visibility of GUI components in a single user interface screen of an application, the visibility manager data structure comprising a mapping table, one or more profiles and a user configuration identifying which of the one or more profiles is to be applied; [[and]]

provide a user interface of a visibility manager via which to receive for the one or more profiles input of respective selections of visibility states of at least a subset of the GUI components; and ~~, the visibility of the GUI components determined in accordance with the visibility state selections of the identified profile~~

display the user interface screen of the application according to an applied one of the profiles using the application and the visibility manager, the displaying of the user interface screen including:

the application building the user interface screen of the application with all of the GUI components set as visible and subsequently calling the visibility manager;

the visibility manager, responsive to the calling:

determining, based on the selected visibility states of the applied profile, which of the GUI components of the built interface screen are to be set as not visible; and

revising the built user interface screen based on the determination; and

the application displaying the revised user interface screen of the application.

18. (Canceled).

19. (Currently Amended) The system as recited in claim 17, wherein

the visibility manager to ~~determine which GUI components to display, and the visibility manager further to~~:

read the one or more profiles upon initialization of the application;

process the one or more profiles;

read and process the user configuration based on the one or more profiles; and

activate the identified profile of the one or more profiles based upon the mapping table.

20. (Canceled).

21. (Previously Presented) The system as recited in claim 19 wherein the visibility manager to:

select an identification of a particular GUI component;

locate the identification in the mapping table;

check a state of a specific GUI component in the application;

compare the state to the identified profile of the one or more of profiles; and

change the state if not in agreement with the identified profile of the one or more profiles.

22. (Previously Presented) The system as recited in claim 21 wherein the state is visible or not visible.

23. (Previously Presented) The system as recited in claim 17 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to objects of an object-oriented and platform independent programming language.

24. (Currently Amended) The system as recited in claim [[18]] 17 wherein

the visibility manager to ~~determine which GUI components to display, and the visibility manager further to~~:

read the one or more profiles upon initialization of the application;

process the one or more profiles;

read and process the user configuration based on the one or more profiles;

activate the identified profile of the one or more profiles based upon the mapping table by:

select an identification of a particular GUI component,

locate the identification in the mapping table,

check a state of a specific GUI component in the application,

compare the state to the identified profile of the one or more of profiles, and

change the state if not in agreement with the identified profile of the one or more profiles~~; revise the user interface based on the activated profile, and display a user interface~~.

25. (Currently Amended) A hardware-implemented computer-readable medium embodying instructions, which, when executed by a processor, cause the processor to perform a method, the method comprising:

providing a user interface of a visibility manager via which to receive, for one or more profiles, input of respective selections of visibility states of at least a subset of GUI components; and

for displaying a user interface screen of the application according to an applied one of the plurality of profiles:

initializing an application, wherein the initializing includes:

starting the application; and

building the user interface screen of the application with all of the GUI components set as visible;

invoking a visibility manager to:

determine, based on the selected visibility states of the applied profile, which of the GUI components of the built user interface screen are to be set as not visible; and

revise the built user interface screen based on the determination; and

displaying [[a]] the revised user interface screen of the application ~~wherein display of the GUI components are determined by the visibility manager based on the visibility state selections~~.

26. (Currently Amended) The medium as recited in claim 25 wherein invoking the visibility manager comprises:

reading the one or more profiles;

processing the one or more profiles;

reading and processing a user configuration based on the one or more profiles; and

activating a particular profile of the one or more profiles~~; revising the user interface based on the particular profile of the one or more profiles; and displaying the user interface~~.

27. (Canceled).

28. (Previously Presented) The medium as recited in claim 26 wherein activating the particular profile of the one or more profiles comprises:

selecting an identification of a particular GUI component;

locating the identification in a mapping table;

checking a state of the particular GUI component;

comparing the state to the particular profile of the one or more profiles; and

changing the state if not in agreement with the particular profile of the one or more profiles.

29. (Original) The medium as recited in claim 26 further comprising:

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

30. (Original) The medium as recited in claim 28 wherein the state is visible or not visible.

31. (Previously Presented) The medium as recited in claim 28 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to objects of an object-oriented and platform independent programming language.

32. (Currently Amended) The medium as recited in claim 25 further comprising:

invoking the visibility manager by:

reading the one or more profiles,

processing the one or more profiles,

reading and processing a user configuration based on the one or more profiles, and

activating a particular profile of the one or more profiles by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table,

checking a state of the particular GUI component,

changing the state if not in agreement with the particular profile of the one or more profiles, and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components~~, revising the user interface based on the particular profile of the one or more profiles, and displaying the user interface; and initializing the application by: starting the application, building the user interface with all of the GUI components visible, and calling the visibility manager after the profile is activated~~.

33. (Currently Amended) A system for managing visibility of GUI components in an application comprising:

first means for interfacing with a user, the means for interfacing providing the GUI components for display; and

means for determining which GUI components are visible, the means for determining including second means for interfacing with a user, the second means for interfacing providing for receipt, for one or more profiles, of respective user selections of visibility states of at least a subset of the GUI components; ~~, the selections used for the determination of which GUI components are visible~~

wherein, for display of a user interface screen of the application according to an applied one of the profiles:

the first means for interfacing is configured to initially build the user interface with all of the GUI components set as visible and subsequently call the means for determining;

the means for determining is configured to, in response to the call:

determine, based on the selected visibility states of the at least a subset of the GUI components, which of the GUI components of the built user interface screen are to be set as not visible;

revise the built user interface screen based on the determination; and

provide the revised user interface screen to the user interface module; and

the first means is configured to display the revised user interface screen.

34. (Previously Presented) The system as recited in claim 33 wherein the means for determining includes:

means for reading the one or more profiles upon initialization of the application;

means for processing the one or more profiles;

means for reading and processing a user configuration based on the one or more profiles; and

means for activating a particular profile of the one or more profiles.

35. (Canceled).

36. (New) A method for displaying a user interface, comprising:

starting up a visibility manager;

an application building a data structure representing a user interface including a plurality of interface components;

during the building of the data structure and responsive to the start-up of the visibility manager, the visibility manager selecting one of a plurality of interface profiles, each of the interface profiles indicating for each of a plurality of interface components a respective visibility instruction;

subsequent to the building of the data structure, the application calling the visibility manager;

responsive to the call, the visibility manager modifying the data structure in accordance with the visibility instructions of the selected interface profile, the modified data structure representing a modified version of the user interface which does not include all of the plurality of interface components; and

the application displaying the modified version of the user interface.

37. (New) The method of claim 36, wherein the starting up of the visibility manager and the building of the data structure by the application are performed in response to a start-up of the application.

38. (New) The method of claim 36, wherein the selection of the selected interface profile is based on a user type associated with a present use of the application.

39. (New) The method of claim 36, wherein the selection of the selected interface profile is based on a user configuration file which identifies which of the plurality of interface profiles a user associated with a present use of the application has set as active.